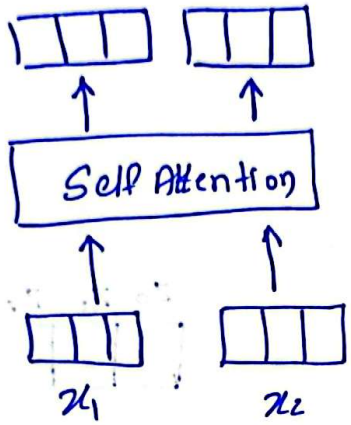
—> Positional Encoding

└> Representing Order of Sequence

① Lion kills tiger

② Tiger kills lion



Words token can be processed parallely

⇓

Draw back

{Lacks the sequential structure of the words}

> Types of Positionel Encoding

① Sinsusoidal Positional Encoding

② Learned Positional Encoding

- Sinsusodial Posttional Encoding

It uses sine and cos functions of different frequencies to create positional encoding.

Formula

$$P.E(position, 2i) = Sin\left(\frac{position}{10000^{2i/dmodel}}\right)$$

$$P.E(position, 2i+1) = Cos\left(\frac{position}{10000^{2i/dmodel}}\right)$$

position = position

i = dimension

dmodel = dimensionality of embeddings

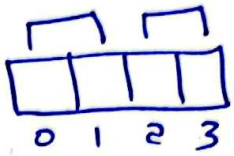
Example

The cat sat

The → [0.1 0.2 0.3 0.4]

cat → [0.5 0.6 0.7 0.8]

sat → [0.9 1.0 1.1 1.2]



$$P.E(pos, 2i) = \sin\left(\frac{pos}{10000^{2i/d_{model}}}\right), \quad P.E(pos, 2i+1) = \cos\left(\frac{pos}{10000^{2i/d_{model}}}\right)$$

d model = 4

for pos = 0

$P.E(0,0) = \sin\left(\frac{0}{10000^{0/4}}\right) = \sin(0) = 0$

$P.E(0,1) = \cos\left(\frac{0}{10000^{0/4}}\right) = \cos(0) = 1$

~~for P.E(0,1)~~

$P.E(0,2) = \sin\left(\frac{0}{10000^{2/4}}\right) = \sin(0) = 0$

$P.E(0,3) = \cos\left(\frac{0}{10000^{2/4}}\right) = \cos(0) = 1$

P.E = [0, 1, 0, 1]

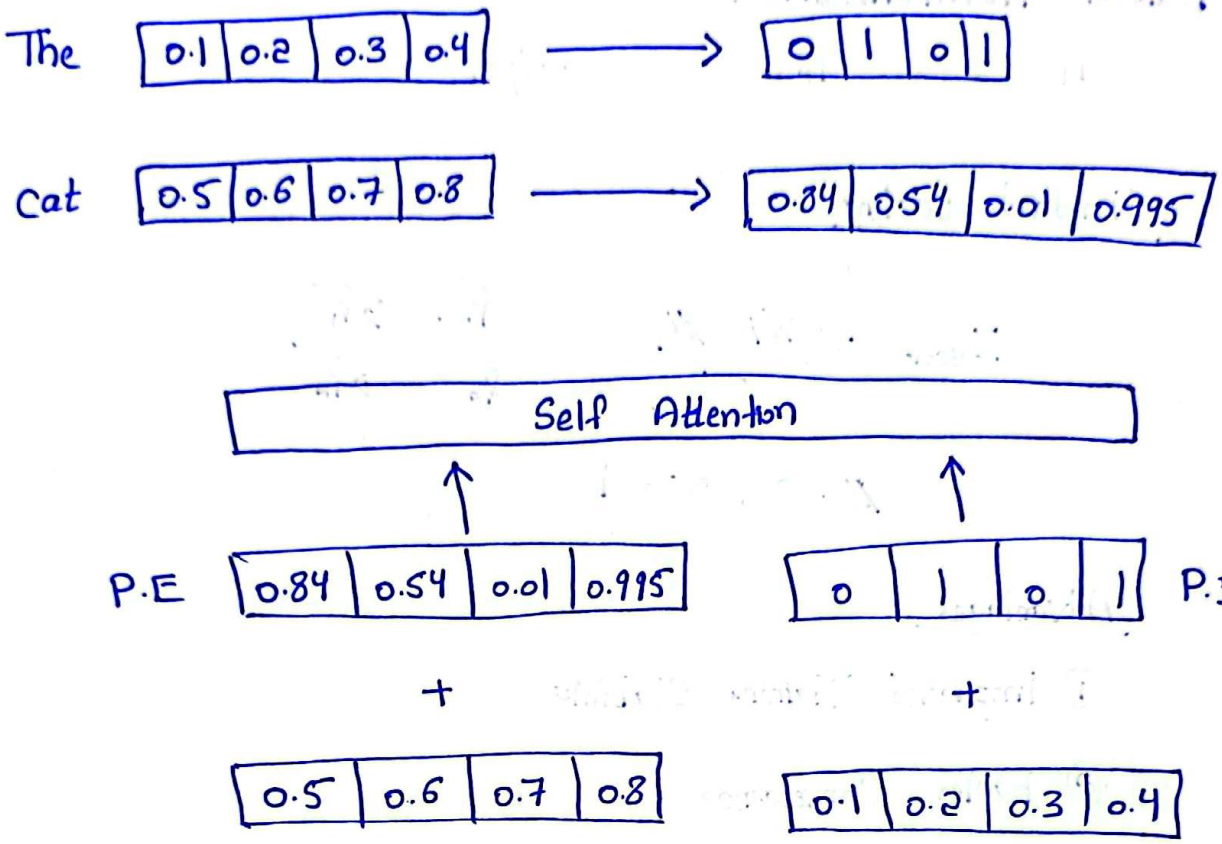
for pos = 1

$P.E(1,0) = \sin\left(\frac{1}{10000^{0/4}}\right) = \sin(1) = 0.8415$

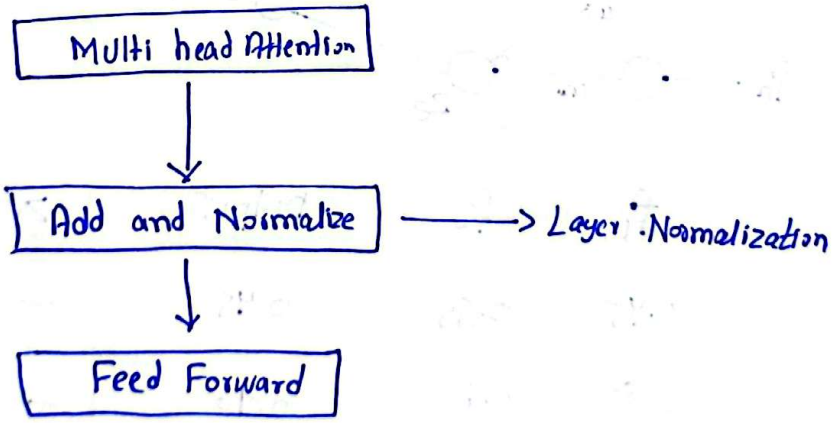
$P.E(1,1) = \cos\left(\frac{1}{10000^{2/4}}\right) = 0.5403$

$P.E(1,2) = \sin\left(\frac{1}{10000^{2/4}}\right) = 0.01$

$P.E(1,3) = \cos\left(\frac{1}{10000^{2/4}}\right) = 0.995$



⟶ Layer Normalization



Normalization

① Batch Normalization

② Layer Normalization

> Batch Normalization

f_1 f_2 output

Standard Scaling

$$Z_{score} = \frac{x_i - \mu}{\sigma}$$

$f_1 \rightarrow f_1'$

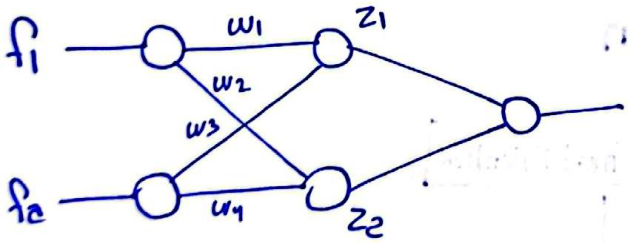
$f_2 \rightarrow f_2'$

$\mu = 0$, $\sigma = 1$

Advantages

① Improved Training Stability

② Faster Convergence

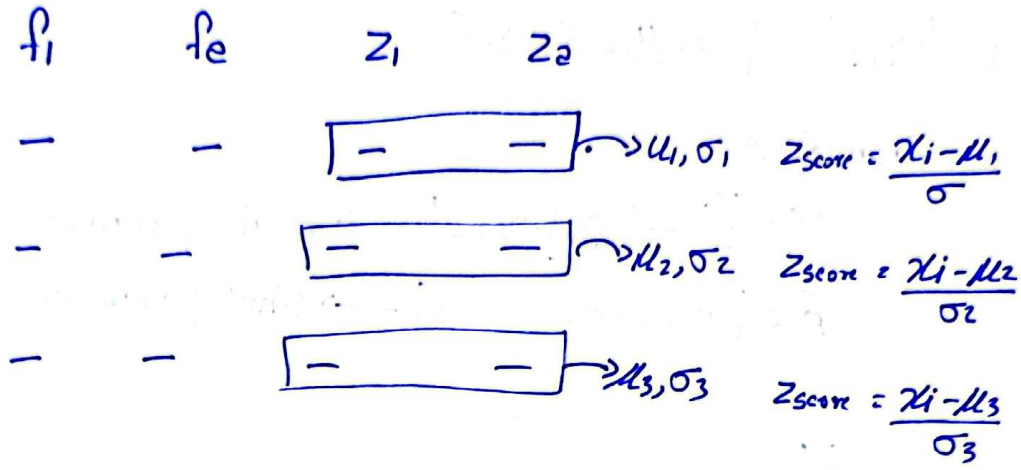


f_1	f_2	Output	z_1	z_2
0.45	0.55	0.45	—	—
0.60	0.20	0.90	—	—

can have different distribution

This can be problem, to solve this, we need to perform normalization on z_1 and z_2

> Layer Normalization



γ, β ⟶ Learnable parameters

Scale and shift parameters

When we do not want to normalize

$$z_1 = \sigma\left[w_1^T x + b_1\right]$$

$$y = \gamma\left[\frac{z_1 - \mu_1}{\sigma}\right] + \beta$$

> Example of Layer Normalization

① "CAT" = [2.0, 4.0, 6.0, 8.0]

② Parameters

$\gamma = [1.0, 1.0, 1.0, 1.0]$ ⟶ Scale paramter

$\beta = [0.0, 0.0, 0.0, 0.0]$ ⟶ Shift parameter

$$Z = \frac{x_i - \mu}{\sigma}$$

Solution

① Compute mean

$$\mu = \frac{2+4+6+8}{4} = \frac{20}{4} = 5.0$$

② Compute standard deviation

$$\sigma^2 = \frac{1}{4}\left[(2-5)^2 + (4-5)^2 + (6-5)^2 + (8-5)^2\right] = \frac{20}{4} = 5$$

③ Normalize Inputs

$$\hat{x}_i = \frac{x_i - \mu}{\sqrt{\sigma^2 + \epsilon}} \qquad \epsilon = 1e^{-5} \Rightarrow \text{to avoid division by 0}$$

$$\sqrt{\sigma^2 + \epsilon} = \sqrt{5 + 1e^{-5}} = \sqrt{5.0001} = 2.236$$

$$\hat{x}_1 = \frac{2-5}{2.236} \approx -1.34$$

$$\hat{x}_2 = \frac{4-5}{2.236} \approx -0.45$$

$$\hat{x}_3 = \frac{6-5}{2.236} \approx 0.45$$

$$\hat{x}_4 = \frac{8-5}{2.236} \approx 1.34$$

Normalized Vector

$$\hat{x}_4 = [-1.34, -0.45, 0.45, 1.34]$$

④ Scale and Shift

$$y_i = \gamma_i \hat{x}_i + \beta_i$$

$$y = [-1.34, -0.45, 0.45, 1.34]$$